User Manual For NewFaces System

1. System Overview

Project Description

NewsFaces is an intelligent face recognition and news analysis system that processes web archive files, extracts articles and images, performs face detection and recognition, and provides comprehensive analytics through a web dashboard.

System Purpose

- **Primary:** Automate face recognition in news articles and images
- **Secondary:** Provide sentiment analysis and content categorization
- **Tertiary:** Generate analytics matrices for system performance evaluation

2. System Architecture

High-Level Components

1. **Data Input Layer:** WARC files, LFW dataset, Wikipedia pages
2. **Processing Layer:** 4-phase pipeline (WARC → Text → Face Enrollment → Face Detection)
3. **Storage Layer:** SQLite database, file system, JSON mappings
4. **Presentation Layer:** Streamlit dashboard with analytics

Technology Stack

- **Backend:** Python 3.8+
- **Face Recognition:** dlib + face_recognition
- **Web Framework:** Streamlit
- **Database:** SQLite
- **Data Processing:** pandas, numpy, scikit-learn
- **Web Scraping:** requests, BeautifulSoup

3. Installation and Setup

Prerequisites

```
# System Requirements
- Python 3.8 or higher
- 8GB RAM minimum
- 10GB free disk space
- Windows 10/11 or Linux

# Python Dependencies
- pip
- virtualenv support
```

Installation Steps

```
# Clone Repository
git clone <repository-url>
```

```
cd Newsfaces-main

# Create Virtual Environment
python -m venv .venv

# Activate Virtual Environment
# Windows:
.venv\Scripts\activate
# Linux/Mac:
source .venv/bin/activate

# Install Dependencies
pip install -r requirements.txt

# Verify Installation
python -c "import face_recognition; print('Installation successful')"
```

Configuration

```
# config/settings.py
LFW_DATASET_PATH = "data/datasets/lfw"
DATABASE_PATH = "data/database/newsfaces.db"
EXTRACTED_DATA_PATH = "data/extracted_data"
LOG_LEVEL = "INFO"
```

4. System Operation

Phase 1: WARC Processing

```
python -c "from src.phases.phase1 import Phase1; Phase1().run()"
```

- Extracts WARC files from `data/warc_files/`
- Parses HTML content and downloads images
- Generates `mappings.json` for content relationships
- Saves extracted data to `data/extracted_data/`

Phase 2: Text Processing

```
python -c "from src.phases.phase2 import Phase2; Phase2().run()"
```

- Processes extracted HTML files
- Performs sentiment analysis
- Extracts named entities
- Categorizes content by topic
- Updates database with analysis results

Phase 3: Face Enrollment

```
python -c "from src.phases.phase3 import Phase3; Phase3().run()"
```

- Loads LFW dataset

- Extracts face encodings (128-dimensional vectors)
- Stores encodings in `known_faces` table
- Prepares face recognition system

Phase 4: Face Detection

```
python -c "from src.phases.phase4 import Phase4; Phase4().run()"
```

- Processes all images in database
- Detects faces in each image
- Recognizes faces against known faces
- Updates face counts and recognition results

5. Dashboard Usage

Starting the Dashboard

```
cd apps
streamlit run streamlit_app_simple.py
# Access at http://localhost:8501
```

Dashboard Features

- **Main Dashboard:** Key metrics, recent activity, articles, face analytics
- **Articles Page:** Search and filter articles, view details
- **Images & Faces Page:** View images, detected faces, re-process images
- **Known Faces Page:** Manage enrolled faces, add new faces
- **Analytics Page:** Distance, similarity, recognition accuracy matrices
- **Search Page:** Advanced multi-type search with filters

6. Data Management

Database Structure

```
-- Main Tables
articles: News articles with metadata
images: Extracted images with face counts
known_faces: Enrolled face encodings
face_recognition_history: Recognition results and performance

-- Key Fields
articles: title, target_uri, sentiment_score, topic_category
images: image_path, face_count, detected_faces
known_faces: name, encoding, confidence_score
```

File Organization

```
data/
├── warc_files/
├── datasets/
│   └── lfw/
```

```
├── extracted_data/
│   ├── html/
│   ├── images/
│   ├── mappings.json
│   └── face_matrices.json
└── database/
    └── newsfaces.db
```

Data Backup

```
cp data/database/newsfaces.db backup/newsfaces_$(date +%Y%m%d).db
tar -czf backup/extracted_data_$(date +%Y%m%d).tar.gz data/extracted_data/
```

7. Troubleshooting

Common Issues

Face Recognition Not Working

```
python -c "import face_recognition"
pip install dlib face_recognition
ls data/datasets/lfw/
```

Database Connection Errors

```
ls -la data/database/
chmod 644 data/database/newsfaces.db
rm data/database/newsfaces.db
python -c "from src.data_access.database import DatabaseManager; DatabaseManager().create_tables()"
```

Image Display Issues

```
ls -la data/extracted_data/images/
chmod 644 data/extracted_data/images/*
```

Memory Issues

```
free -h  # Linux
wmic computersystem get TotalPhysicalMemory  # Windows
# Reduce batch sizes in phase files
```

Performance Optimization

Database Optimization

```
CREATE INDEX idx_articles_title ON articles(title);
CREATE INDEX idx_images_article_id ON images(article_id);
CREATE INDEX idx_known_faces_name ON known_faces(name);
VACUUM;
ANALYZE;
```

Processing Optimization

```
BATCH_SIZE = 100  # Process in batches
import multiprocessing
pool = multiprocessing.Pool(processes=4)
```

8. Maintenance

Regular Tasks

Frequency	Tasks
Daily	Check logs, monitor disk, verify dashboard
Weekly	Database optimization, clean temp files, review performance
Monthly	Full backup, update dependencies, optimize

Log Management

```
tail -f logs/newsfaces.log
logrotate -f /etc/logrotate.d/newsfaces
find logs/ -name "*.log" -mtime +30 -delete
```

9. Security Considerations

- **Local Storage Only**
- **No External Data Transmission**
- **Dashboard Access Control via Streamlit**
- Regular security updates and access logging
- Secure file permissions and encrypted backups

10. Support and Documentation

Getting Help

1. Check logs
2. Read manual and comments
3. Search project repository issues

Useful Commands

```
python -c "from src.data_access.database import DatabaseManager; db=DatabaseManager(); print(f'Articles: {db.get_article_count()}, Images: {db.get_image_count()}, Faces: {db.get_known_faces_count()}')"
python -c "import face_recognition; print('Face recognition available')"
python -c "from src.data_access.database import DatabaseManager; db=DatabaseManager(); print('Database connection successful')"
python -c "from src.services.face_service import FaceService; fs=FaceService(); print('Face service initialized')"
```

Dependency Matrix - NewsFaces System

1. Package Dependencies

Core Dependencies

Package	Version	Purpose	Required
Python	3.8+	Runtime environment	✅
face_recognition	1.3.0+	Face detection/recognition	✅
dlib	19.22+	Face processing backend	✅
streamlit	1.28+	Web dashboard	✅
pandas	1.5+	Data manipulation	✅
numpy	1.21+	Numerical computing	✅
Pillow	9.0+	Image processing	✅
requests	2.28+	HTTP requests	✅
beautifulsoup4	4.11+	HTML parsing	✅
plotly	5.0+	Data visualization	✅

Optional Dependencies

Package	Version	Purpose	Required
scikit-learn	1.1+	Machine learning	❌
transformers	4.20+	NLP processing	❌
opencv-python	4.6+	Computer vision	❌
warcio	1.7+	WARC file processing	❌

2. System Dependencies

Operating System

OS	Version	Support	Notes
Windows	10/11	✅ Full	Tested
Linux	Ubuntu 18.04+	✅ Full	Recommended for production
macOS	10.15+	⚠️ Partial	Possible dlib build issues

Hardware Requirements

Component	Minimum	Recommended	Notes
CPU	4 cores	8+ cores	Multi-core for parallelism
RAM	8GB	16GB+	Large datasets
Storage	10GB	50GB+	WARC files and images
GPU	None	NVIDIA GPU	Optional for speedups

3. File Dependencies

Input Files

File Type	Location	Purpose	Required
WARC files	data/warc_files/	Web archive data	✅
LFW dataset	data/datasets/lfw/	Face training	✅
Config	config/settings.py	System settings	✅

Generated Files

File Type	Location	Purpose	Generated By
Database	data/database/newsfaces.db	Data storage	DatabaseManager
Mappings	data/extracted_data/mappings.json	Content relationships	WARCService

File Type	Location	Purpose	Generated By
Matrices	data/extracted_data/face_matrices.json	Analytics data	Analytics Engine
Logs	logs/newsfaces.log	System logs	LoggingUtils

4. Service Dependencies

Service Hierarchy

```
Streamlit Dashboard
  ↓
FaceService → FaceProcessor → face_recognition library
  ↓
TextService → TextProcessor → NLP libraries
  ↓
WARCService → FileManager → OS file system
  ↓
DatabaseManager →
```